

# Analiza obiektowa

3@KASK

12 czerwca 2009

Symbol projektu: 3@KASK	Opiekun projektu: mgr inż. Tomasz Boiński
Nazwa Projektu: Wizualizacja grafów za pomocą biblioteki Prefuse	

Nazwa Dokumentu: Analiza obiektowa	Nr wersji: 0.0
Odpowiedzialny za dokument: Piotr Kunowski	Data pierwszego sporządzenia: 23 maja 2009
Przeznaczenie: DLA KLIENTA	Data ostatniej aktualizacji: 12 czerwca 2009

## Historia dokumentu

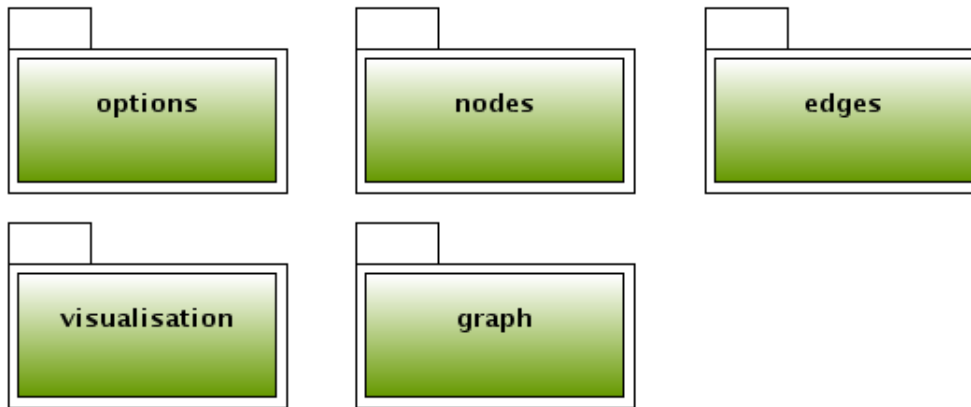
Wersja	Opis modyfikacji	Rozdział/strona	Autor modyfikacji	Data
1	Stworzenie	wszystkie	Grupa projektowa	23.05.09

## Spis treści

<b>1</b>	<b>Pakiety</b>	<b>3</b>
1.1	Diagram . . . . .	3
1.2	Opis pakietów . . . . .	3
<b>2</b>	<b>Pakiet options</b>	<b>4</b>
2.1	Diagram . . . . .	4
2.2	Opis klasy . . . . .	4
<b>3</b>	<b>Pakiet nodes</b>	<b>7</b>
3.1	Diagram . . . . .	7
3.2	Opis klasy . . . . .	7
<b>4</b>	<b>Pakiet edges</b>	<b>12</b>
4.1	Diagram . . . . .	12
4.2	Opis klasy . . . . .	12
<b>5</b>	<b>Pakiet visualization</b>	<b>15</b>
5.1	Diagram . . . . .	15
5.2	Opis klasy . . . . .	15
<b>6</b>	<b>Pakiet graph</b>	<b>16</b>
6.1	Diagram . . . . .	16
6.2	Opis klasy . . . . .	17
<b>7</b>	<b>Pakiet utils</b>	<b>17</b>
7.1	Diagram . . . . .	17
7.2	Opis klasy . . . . .	17

# 1 Pakiety

## 1.1 Diagram



## 1.2 Opis pakietów

P001	options
Opis:	Pakiet zawierający klasy z polami opisującymi różne (modyfikowalne) ustawienia wizualizacji takie jak: kolory, grubość linii itp.
Interfejsy:	
Realizowane wymagania:	WF002, WF001, WI004
Priorytet:	średnio ważne

P002	nodes
Opis:	Pakiet z klasami odpowiedzialnymi za wizualizację i przechowywanie danych o wierzchołkach.
Interfejsy:	
Realizowane wymagania:	WF004, WF005, WF006, WF007, WI004
Priorytet:	bardzo ważne

P003	edges
Opis:	Pakiet z klasami odpowiedzialnymi za wizualizację i przechowywanie danych o krawędziach.
Interfejsy:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	bardzo ważne


P004	visualization
Opis:	Zawiera dodatkowe klasy przydatne w wizualizacji.
Interfejsy:	
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	średnio ważne


P005	graph
Opis:	Pakiet zawiera klasy, które zawierają podstawowe operacje na danych OwlApi oraz graph.
Interfejsy:	
Realizowane wymagania:	WD001
Priorytet:	bardzo ważne

P005	utils
Opis:	Pakiet zawiera klasy pomocnicze
Interfejsy:	
Realizowane wymagania:	CF005
Priorytet:	bardzo ważne

## 2 Pakiet options

### 2.1 Diagram

 NodeColors
<i>Attributes</i>
protected Color allValuesFromNodeColor
protected Color anonymousClassNodeColor
protected Color cardinalityNodeColor
protected Color cardinalityValueNodeColor
protected Color classNodeColor
protected Color complementOfNodeColor
protected Color dataTypeNodeColor
protected Color differentNodeColor
protected Color functionalPropertyNodeColor
protected Color individualNodeColor
protected Color informationNodeColor
protected Color intersectionOfNodeColor
protected Color inverseFunctionalPropertyColor
protected Color maxCardinalityValueNodeColor
protected Color minCardinalityValueNodeColor
protected Color nothingNodeColor
protected Color oneOfNodeColor
protected Color propertyNodeColor
protected Color sameAsNodeColor
protected Color someValuesFromNodeColor
protected Color symmetricPropertyNodeColor
protected Color thingNodeColor
protected Color transitivePropertyNodeColor
protected Color unionOfNodeColor

 EdgeColors
<i>Attributes</i>
protected Color rangeEdgeColor
protected Color domainEdgeColor
protected Color edgeColor
protected Color equivalentEdgeColor
protected Color equivalentPropertyEdgeColor
protected Color functionalEdgeColor
protected Color inverseOfEdgeColor
protected Color propertyEdgeColor
protected Color subEdgeColor

### 2.2 Opis klasy

CO001	EdgeColors
Opis:	Zawiera definicje kolorów dla poszczególnych rodzajów krawędzi.

Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> <li>• domainEdgeColor</li> <li>• edgeColor</li> <li>• equivalentEdgeColor</li> <li>• equivalentPropertyEdgeColor</li> <li>• functionalEdgeColor</li> <li>• inverseOfEdgeColor</li> <li>• propertyEdgeColor</li> <li>• rangeEdgeColor</li> <li>• subEdgeColor</li> </ul>
Metody:	
Realizowane wymagania:	WF002
Priorytet:	średnio ważny
CO002	NodeColors
Opis:	Zawiera definicje kolorów dla poszczególnych rodzajów krawędzi.
Klasy nadrzędne:	

Atrybuty:	<ul style="list-style-type: none"> <li>• allValuesFromNodeColor</li> <li>• cardinalityNodeColor</li> <li>• cardinalityValueNodeColor</li> <li>• classNodeColor</li> <li>• complementOfNodeColor</li> <li>• dataTypeNodeColor</li> <li>• differentNodeColor</li> <li>• functionalPropertyNodeColor</li> <li>• individualNodeColor</li> <li>• informationNodeColor</li> <li>• intersectionOfNodeColor</li> <li>• inverseFunctionalNodeColor</li> <li>• maxCardinalityValueNodeColor</li> <li>• minCardinalityValueNodeColor</li> <li>• nothingNodeColor</li> <li>• oneOfNodeColor</li> <li>• propertyNodeColor</li> <li>• sameAsNodeColor</li> <li>• someValuesFromNodeColor</li> <li>• symmetricPropertyNodeColor</li> <li>• thingNodeColor</li> <li>• transitivePropertyNodeColor</li> <li>• unionOfNodeColor</li> </ul>
Metody:	
Realizowane wymagania:	WF002
Priorytet:	średnio ważny



Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF006, WF007, WI004
Priorytet:	ważne

CN003	AnonymousClassNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WI004
Priorytet:	ważne

CN004	CardinalityNode
Opis:	
Klasy nadrzędne:	AnonymousNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN005	CardinalityValueNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN006	ClassNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF005, WI004
Priorytet:	ważne

CN007	ComplementOfNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN008	DatatypeNode
Opis:	



Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WI04
Priorytet:	ważne

CN009	DifferentNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN010	FunctionalPropertyNode
Opis:	
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN011	IndividualNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WI004
Priorytet:	ważne

CN012	InformationNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF010, WI004
Priorytet:	ważne

CN013	IntersectionOfNode
Opis:	
Klasy nadrzędne:	AnonymousNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WI004
Priorytet:	ważne

CN014	inverseFunciotnalPropertyNode
Opis:	

Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN015	MaxCardinalityValueNode
Opis:	
Klasy nadrzędne:	CardinalityValueNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN016	MinCardinalityValueNode
Opis:	
Klasy nadrzędne:	CardinalityValueNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN017	NothingNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF005, WI004
Priorytet:	ważne

CN018	OneOfNode
Opis:	
Klasy nadrzędne:	AnonymousClassNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

CN019	PropertyNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF007, WI004
Priorytet:	ważne

CN020	SameAsNode
Opis:	

Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

CN021	SomeValuesFromPropertyNode
Opis:	
Klasy nadrzędne:	PropertyNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

CN022	SymmetricPropertNode
Opis:	
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

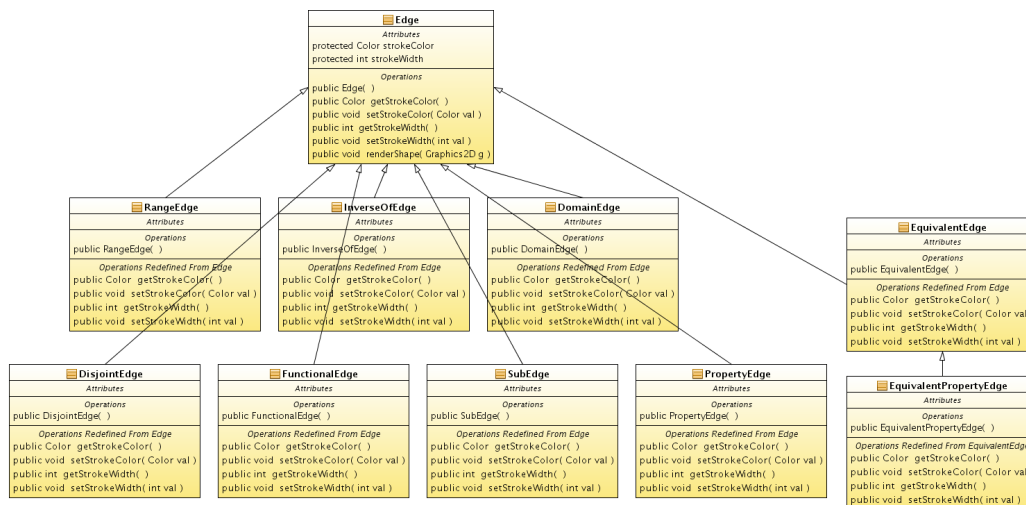
CN023	ThingNode
Opis:	
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF005, WI004
Priorytet:	ważne

CN024	TreansitivePropertyNode
Opis:	
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN025	UnionOfNode
Opis:	
Klasy nadrzędne:	AnonymousNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

## 4 Pakiet edges

### 4.1 Diagram



### 4.2 Opis klasy

CE001	Edge
Opis:	Klasa reprezentująca krawędź na grafie
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> <li>• Color strokeColor</li> <li>• int strokeWidth</li> <li>• boolean hasArrow</li> <li>• boolean hasInvertedArrow</li> <li>• Polygon arrowHead</li> <li>• Color arrowHeadColor</li> </ul>

Metody:	<ul style="list-style-type: none"> <li>• <code>getStrokeColor ()</code></li> <li>• <code>setStrokeColor (Color val)</code></li> <li>• <code>getStrokeWidth ()</code></li> <li>• <code>setStrokeWidth (int val)</code></li> <li>• <code>getArrowHead()</code></li> <li>• <code>setArrowHead(Polygon arrowHead)</code></li> <li>• <code>isHasArrow()</code></li> <li>• <code>setHasArrow(boolean hasArrow)</code></li> <li>• <code>isHasInvertedArrow()</code></li> <li>• <code>setHasInvertedArrow(boolean hasInvertedArrow)</code></li> <li>• <code>getArrowHeadColor()</code></li> <li>• <code>setArrowHeadColor(Color arrowHeadColor)</code></li> </ul>
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	bardzo ważne

CE002	DisjointEdge
Opis:	
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE003	DomainEdge
Opis:	
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE004	EquivalentEdge
Opis:	
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE005	EquivalentPropertyEdge
Opis:	

Klasy nadrzędne:	EquivalentEdge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE006	FunctionalEdge
Opis:	
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE007	InverseOfEdge
Opis:	
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

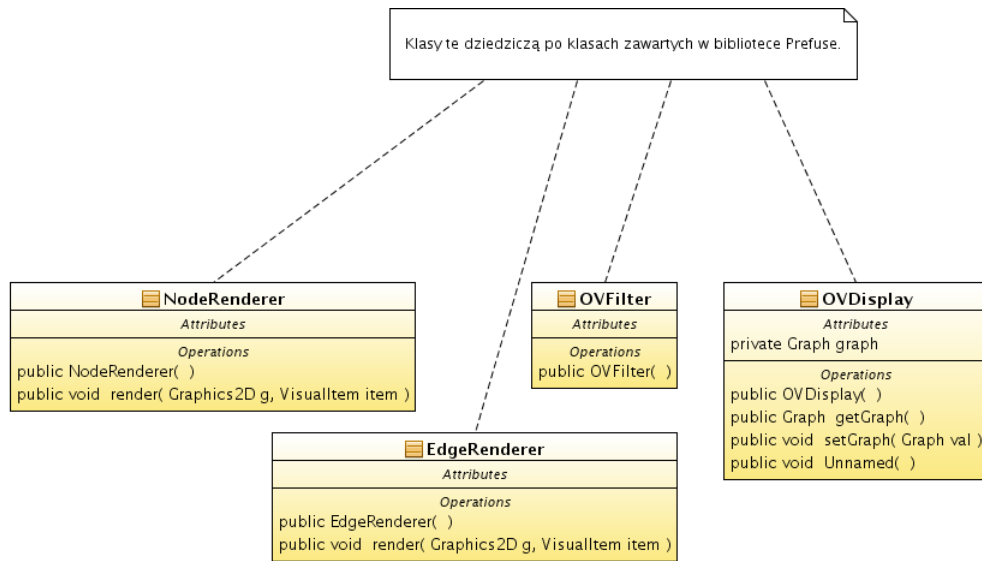
CE008	PropertyEdge
Opis:	Klasa reprezentująca na grafie relację między property a klasą
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE009	RangeEdge
Opis:	Klasa reprezentująca na grafie właściwość property Range
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE010	SubEdge
Opis:	Klasa reprezentująca na grafie związek SubClass pomiędzy klasami
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

## 5 Pakiet visualization

### 5.1 Diagram



### 5.2 Opis klasy

CV001	EdgeRenderer
Opis:	Klasa przeciążająca metody renderowania krawędzi grafu z biblioteki prefuse.
Klasy nadrzędne:	prefuse.render.EdgeRenderer
Atrybuty:	
Metody:	<ul style="list-style-type: none"> <li>render(Graphics2D g, VisualItem item) - metoda renderująca krawędź</li> </ul>
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	ważne

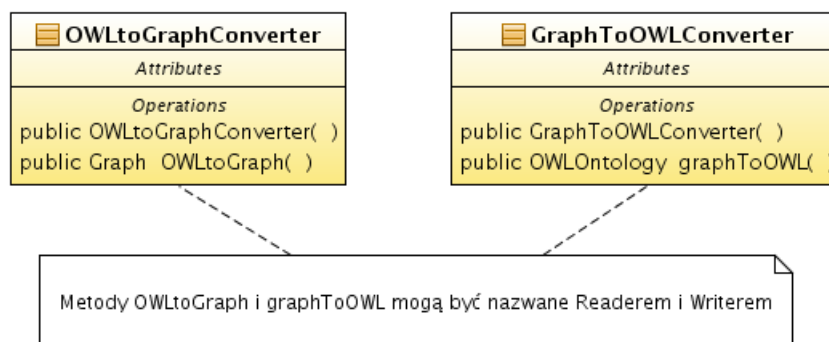
CV002	NodeRenderer
Opis:	Klasa przeciążająca metody renderowania wierzchołków grafu z biblioteki prefuse.
Klasy nadrzędne:	prefuse.render.LabelRenderer
Atrybuty:	
Metody:	<ul style="list-style-type: none"> <li>render (Graphics2D g, VisualItem item) - metoda renderująca wierzchołek</li> <li>drawString(Graphics2D g, FontMetrics fm, String text, boolean useInt, double x, double y, double w) - metoda wypisująca na wierzchołku String</li> </ul>
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	ważne

CV003	OVDisplay
Opis:	Klasa tworząca obiekt JComponent do umieszczenia na okienku JAVA zawierający wygenerowany graf z wizualizacją
Klasy nadrzędne:	prefuse.Display
Atrybuty:	<ul style="list-style-type: none"> <li>Graph graph - obiekt typu prefuse.data.graph zawierający dane o grafie do wyświetlenia.</li> </ul>
Metody:	<ul style="list-style-type: none"> <li>getGraph() - zwraca graf z wyświetlanymi danymi</li> <li>setGraph(Graph graph) - nadpisuje obecny graf podanym</li> <li>generateGraphFromOWL(OWLontology ont) - wpisuje do klasy obiekt Grpah wygenerowany na podstawie ontologii</li> </ul>
Realizowane wymagania:	WF001, WF002, WF008, WI004
Priorytet:	ważne

CV004	OVFilter
Opis:	Klasa zawierająca filtry służące do wyświetlania danych w różnych zakresach
Klasy nadrzędne:	
Atrybuty:	
Metody:	
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	ważne

## 6 Pakiet graph

### 6.1 Diagram





## 6.2 Opis klasy

CG001	GraphToOWLConverter
Opis:	Klasa zawierająca metody pozwalające na przetwarzanie obiektów grafów z prefuse na obiekty OWL API. Klasa jest singletonem.
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> <li>• INSTANCE - instancja klasy GraphToOWLConverter</li> </ul>
Metody:	<ul style="list-style-type: none"> <li>• getInstance() - zwraca instancję klasy</li> <li>• GraphToOWL(OWLontology ontology) -Zamienia graf z biblioteki prefuse na ontologię zapisana w OWL API.</li> </ul>
Realizowane wymagania:	WD001, WI004
Priorytet:	ważne

CG002	OWLtoGraphConverter
Opis:	Klasa zawierająca metody pozwalające na przetwarzanie obiektów OWL API na obiekty prefuse. Klasa jest singletonem.
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> <li>• INSTANCE - instancja klasy GraphToOWLConverter</li> </ul>
Metody:	<ul style="list-style-type: none"> <li>• getInstance() - zwraca instancję klasy</li> <li>• recursiveSubClassReader(Node parent, OWLClass cls, OWLontology ontology ) - wczytuje do grafu OWL wszystkie klasy wraz z ich podklasami.</li> <li>• OWLtoGraph(OWLontology ontology) -Zamienia ontologię w OWL API na graf z biblioteki prefuse.</li> </ul>
Realizowane wymagania:	WD001, WI004
Priorytet:	ważne

## 7 Pakiet utils

### 7.1 Diagram

### 7.2 Opis klasy

CU001	Debug
Opis:	Klasa do użycia przy debugowaniu, zapewnia strumień z błędami zwracanymi przez bibliotekę. Klasa jest singletonem.

Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"><li>• INSTANCE - instancja klasy Debug</li><li>• Debug - Strumień do którego wpisywane są informacje potrzebne do debugowania</li></ul>
Metody:	<ul style="list-style-type: none"><li>• getInstance() - zwraca instancję klasy</li><li>• setStream(PrintStream ps) - ustawia podany strumień jako strumień na który zwracane będą błędy</li><li>• sendMessage(String s) - wysyła wiadomość na strumień do debugowania, jeżeli został wcześniej podpięty za pomocą funkcji setStream</li></ul>
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	bardzo ważne